２－１０：関数

関数とは複数の命令をひとつにまとめたようなものです。

プログラムを作っていくと、何度も同じ処理を繰り返す場合が多々あります。そのたびに同じ命令を何度も書いていくのは大変面倒で、プログラム文自体も見にくくなります。

そこで関数を使えば、重複する処理をたった一文でかくことができます。

● 関数の形の種類

関数を使うときは変数のように関数の型を宣言する必要があります。型の種類は返り値を返す「int 型」や「bool 型」、返り値を返さない「void 型」があります。整数を返す必要があるなら「int 型」、真偽を返すなら「bool 型」、返り値が必要ない場合は「void 型」を使いましょう。

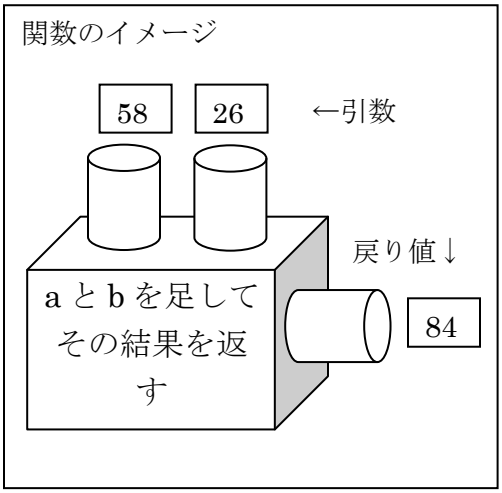
また返り値を返す場合の型は、int 型、bool 型や float 型だけでなく、構造体でも可能です。構造体についてはこの資料の２－１２をごらんください。

● 引数と戻り値

関数を扱う上で、引数と戻り値という言葉があります。入力する値が引数、関数から返す値が戻り値となります。

関数が値を返すときは、関数の中に「return 返す値;」と書けばその値を返すことができるようになります。

また引数の数はいくつでも入力でき、場合によっては引数無しの関数も作成できます。



● 関数の宣言と呼び出し

引数と戻り値がある関数をここで作ってみましょう。ここでは「a と b を足してその結果を返り値として返す」という関数「Add」をつくってみます。

まずはじめに、ヘッダーファイルのユーザー宣言（「private:」または「public:」のところに）下記のように宣言します。

```
int __fastcall Add(int a, int b);
```

※ fastcall の前の線はアンダーバー2 個（__fastcall は別に書かなくてもいい）

そして処理の内容を cpp ファイルに書きます。これは自分の好きな場所にかけます。

```
//---------------------------------------------------------
int __fastcall TForm1::Add(int a, int b)
{
    int z;
    z = a + b;
    return z;
}
//---------------------------------------------------------
```

これで関数 Add がつかえるようになりました。関数の処理を実行したいときは、使いたいところで「関数名(引数)」と書けば実行されます。

```
Add(58,26);
```

これで 58 と 26 を足した結果が返ってきます。

もしプログラムに整数 z=Add(58,26)と書いたなら、z に 58+26 の 84 が代入されます。

● 引数、返り値のない関数

関数は引数を与えたり、返り値を返さない関数を返すこともできます。例として Shape の性質を一度に変えるという関数を作ってみます。

宣言する場所は先ほどと変わりありません。しかし、今回は引数がないので()の中に何かを書き込む必要はありません。

```
void __fastcall ShapeShokika();
```

処理の内容も先ほどと同じで好きな場所にかけます。

```
//---------------------------------------------------------
void __fastcall TForm1::ShapeShokika()
{
    Shape1->Left = 250;
    Shape1->Top = 200;
    Shape1->Width = 50;
    Shape1->Height = 50;
    Shape1->Brush->Color = clRed;
    Shape1->Shape = stCircle;
}
//---------------------------------------------------------
```

これで ShapeShokika という関数ができました。この関数をよびだすと、Shape1 が指定の位置におかれ、丸く赤くなります。呼び出し方は

```
ShapeShokika();
```

と書いてやるだけで呼び出せます。

関数は大きなプログラムを書いていく上で非常に重要なものです。知らなくてもプログラムはかけないことも無いのですが、とても見にくくなり、面倒な書き損じによるバグが発生しやすくなります。

ですが、関数を使うと、プログラムが見やすくなり、デバッグもとても楽になります。

関数は自分で作るもの以外にも、標準ですでに作られている関数もたくさんあります。いろいろやってみてください。